

Contando submatrices

Contribución de Natalia Pérez y Laura Rivero

Descripción del problema

Próximamente se lanzará al mercado "SigMatriz", un nuevo juego apto para muchos participantes, diseñado por una empresa de sistemas en expansión. Sus reglas son sencillas:

1) Se visualizan por pantalla durante unos pocos segundos dos números **A** y **B** ($0 \leq A \leq B \leq 2.560.000$)

2) Los números desaparecen y se muestra una matriz de **M** x **N** ($1 \leq M \leq 160$; $1 \leq N \leq 160$) en donde cada casilla contiene un número entero **E** ($0 \leq E \leq 100$).

El objetivo del juego es encontrar la cantidad de submatrices de cualquier tamaño **m** x **n** ($1 \leq m \leq M$; $1 \leq n \leq N$) donde la suma **S** de los números de cada casilla cumpla con $A \leq S \leq B$.

Compiten todos los participantes que se encuentren en línea y gana la partida aquel que logre hacer entrar primero al jurado un e-mail con la cantidad correcta.

La empresa de sistemas requiere de tu ayuda escribiendo la función:

```
sigmatriz( A, B, M, N,
P, Matriz, Respuestas,
Tiempo )
```

que retorne el número del participante ganador del juego o 0 en caso de no existir un ganador.

Sus parámetros son:

A : límite inferior de la suma ($0 \leq A \leq B$)

B : límite superior de la suma ($A \leq B \leq 2.560.000$)

M : cantidad de filas de la matriz ($1 \leq M \leq 160$)

N : cantidad de columnas de la matriz. ($1 \leq N \leq 160$)

Matriz : arreglo de tamaño **M** x **N** con la matriz inicial del juego.

P : cantidad de participantes del juego ($0 \leq P \leq 1000$)

Respuestas : arreglo de tamaño **P** con la cantidad de submatrices que contó cada participante.

En C/C++ `int sigmatriz(int A, int B, int M, int N, int P, int Matriz[][160], int Respuestas[], int Tiempo[])`

Debes incluir también un archivo de encabezamiento **sigmatriz.h**

Tiempo : arreglo de tamaño **P** que indica el orden en que se recibieron los e-mails con las respuestas.

Ejemplo

Si se presenta el siguiente caso:

A= 22; B=27; M= 4; N= 3; P=5;

Matriz:

15	5	8
22	4	1
5	7	2
1	11	6

Respuestas= 7;9;9;8;9

Tiempo= 2;4;3;1;5

La solución deberá ser **3**;

Las submatrices que tuvieron que contabilizar se ilustran a continuación:

15	5	8
22	4	1
5	7	2
1	11	6

15	5	8
22	4	1
5	7	2
1	11	6

15	5	8
22	4	1
5	7	2
1	11	6

15	5	8
22	4	1
5	7	2
1	11	6

15	5	8
22	4	1
5	7	2
1	11	6

15	5	8
22	4	1
5	7	2
1	11	6

15	5	8
22	4	1
5	7	2
1	11	6

15	5	8
22	4	1
5	7	2
1	11	6

15	5	8
22	4	1
5	7	2
1	11	6

Detalles de implementación

En un único archivo, llamado **sigmatriz.cpp**, **sigmatriz.c**, o **sigmatriz.pas**. debes enviar una función que implemente la función descrita antes, usando los siguientes prototipos:

Para programadores Pascal

En Pascal function sigmatriz(A, B, M, N, P : longint ;
Matriz: array[1..160, 1..160] of longint; Respuestas, Tiempo
array[0..999] of longint) : longint

Evaluador local

El evaluador local (programa para probar ejemplos propios) lee la entrada por `stdin` en el siguiente formato:

- Línea 1: A, B
- Línea 2: M, N, P
- Línea 3+i ($0 \leq i < M$) N números separados por blanco indicando la i-ésima fila de la matriz.
- Línea 3+M: P números separados por blanco correspondientes a las `Respuestas[j]` ($0 \leq j < P$)
- Línea 4+M: P números separados por blanco correspondientes a los `Tiempo[j]` ($0 \leq j < P$)

El evaluador entrega el valor de la función por consola.

Para el caso del ejemplo la entrada sería:

22 27
4 3 5
15 5 8
22 4 1
5 7 2
1 11 6
7 9 9 8 9
2 4 3 1 5

Subtareas

28 puntos se asignarán a problemas en los cuales la entrada tenga **M = 1**.

22 puntos adicionales para aquellos problemas en donde se asegure que las submatrices que conforman la respuesta tengan **n=1** y/o **m=1**.

50 puntos adicionales resolviendo el enunciado completo.